A Robust Access Control Protocol for the Smart Grid Systems

Muhammad Tanveer, Abd Ullah Khan, Neeraj Kumar, *Senior Member, IEEE*, Alamgir Naushad, and Shehzad Ashraf Chaudhry

***Abstract*—Lightweight cryptography (LWC)-based authenticated encryption with associative data (AEAD) cryptographic primitives require fewer computational and energy resources than conventional cryptographic primitives as a single operation of an AEAD scheme provides confidentiality, integrity, and authenticity of data. This feature of AEAD schemes helps design an access control (AC) protocol to be leveraged for enhancing the security of the resource-constrained Internet of Things (IoT)-enabled smart grid (SG) system with low computational overhead and fewer cryptographic operations. This article presents a novel and robust AC protocol, called RACP-SG, which aims to enhance the security of resource-constrained IoT-enabled SG systems. RACP-SG employs an LWC-based AEAD scheme, ASCON and the hash function, ASCON-hash, along with elliptic curve cryptography to accomplish the AC phase. Besides, RACP-SG enables a smart meter (SM) and a service provider (SEP) to mutually authenticate each other and establish a session key (SK) while communicating across the public communication channel. By using the SK, the SM can securely transfer the gathered data to the SEP. We verify the security of the SK using the widely accepted random oracle model. Moreover, we conduct Scyther-based and informal security analyses to demonstrate that RACP-SG is protected against various covert security risks, such as replay, impersonation, and desynchronization attacks. Besides, we present a comparative study to illustrate that RACP-SG renders superior security features while reducing energy, storage, communication, and computational overheads compared to the state of the art.**

***Index Terms*—Access control (AC), authenticated encryption with associative data (AEAD), authentication, privacy, security, smart grid (SG).**

Manuscript received April 15, 2021; revised July 20, 2021; accepted September 14, 2021. Date of publication September 17, 2021; date of current version April 25, 2022. *(Corresponding author: Neeraj Kumar.)*

Muhammad Tanveer and Abd Ullah Khan are with the Faculty of Computer Science and Engineering, GIK Institute of Engineering Sciences and Technology, Topi 23640, Pakistan (e-mail: tanveer.m@giki.edu.pk; newabd470@gmail.com).

Neeraj Kumar is with the Department of Computer Science Engineering, Thapar Institute of Engineering and Technology, Deemed University, Patiala 147004, India, also with the School of Computer Science, University of Petroleum and Energy Studies, Dehradun 248007, India, and also with the Department of Computer Science and Information Engineering, Asia University, Taichung City 413, Taiwan (e-mail: neeraj.kumar@thapar.edu).

Alamgir Naushad is with the School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad 44000, Pakistan (e-mail: anaushad@nbc.nust.edu.pk).

Shehzad Ashraf Chaudhry is with the Department of Computer Engineering, Istanbul Gelisim University, 34310 Istanbul, Turkey (e-mail: sashraf@gelisim.edu.tr).

Digital Object Identifier 10.1109/JIOT.2021.3113469

I. Introduction

A CYBER–PHYSICAL system (CPS) contains multiple components that interact and communicate using the public Internet [1], [2]. This way, CPS is envisaged to be an essential part of future applications. Among these applications, smart grid (SG) systems are conceived to be the most important, wherein a user's sensitive information is transmitted from and to the user. Particularly, SG systems integrated with the Internet of Things (IoT) are essential for realizing smart homes. This way, SG systems enable users to customize the power utilization and its cost, thereby leading to smart homes applications [3].

An SG system contains service providers (SEPs) and smart meters (SMs). SEPs perform actuation, control, and communication processes to ensure an uninterruptible and flawless power supply. The SMs contain sensing and communicating modules and are responsible for collecting and transmitting the information to SEPs in real time via public channels [4], [5]. Such channels are prone to various types of attacks that enable an adversary to access the exchanged information between SMs and SEPs in the SG system. Therefore, a secure and robust access control (AC) protocol is necessary to enable the SG system entities to exchange information securely after establishing the session key (SK).

Several authenticated key exchange (AKE) and AC protocols have been proposed that enable the SG system's components to communicate after establishing an SK securely. However, most of them cannot protect users' anonymity and untraceability. Besides, many of them are prone to man-in-the-middle (MITM), replay, SM and SEP impersonation, privilege-insider (PI), ephemeral secret leakage (ESL), and SM physical capture attacks. Additionally, a number of these protocols are unable to provide mutual authentication and SK security. We propose an AC protocol to overcome the security threats and vulnerabilities associated with the existing AKE and AC protocols. The proposed AC protocol uses lightweight cryptography (LWC)-based authenticated encryption with associative data (AEAD) in conjunction with elliptic curve cryptography (ECC).

A. Novelty and Research Contributions

Recently, an increasing number of AEAD algorithms are being proposed, focusing on the provisioning of encryption/decryption functions in resource-constrained devices [6]. As shown in Fig. 1, at the source side, an AEAD scheme

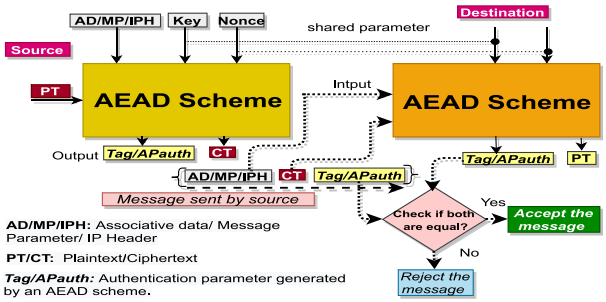

Fig. 1. High-level depiction of AEAD schemes' functionality—the base module of our proposed protocol.

accepts a key, an initialization vector/nonce, associative data (AD), and plaintext (PT) as inputs and generates the ciphertext (CT) and Authentication Parameters (Tag/APauth) as outputs. The AD indicates the data required to be secured in an unencrypted state. For instance, the information contained in an IP header, or any part of a message (identity, pseudo-identity), requiring essential integrity at the destination, can be considered as AD here. The PT in the scheme is made confidential through the CT generated by an AEAD scheme. Similarly, both CT and AD are authenticated by APauth, such that an APauth carries out the message authentication functionality to facilitate the AD and CT authentication at the destination. This approach adopted by an AEAD scheme enables it to ensure data integrity, confidentiality, and authenticity simultaneously and with a single operation. This suggests that an AEAD-based scheme can potentially lead toward a reduced number of cryptographic operations required to be performed in an AC process. Besides, the AEAD scheme is less resource intensive and is suitable for an environment where many devices communicate with the server. Therefore, a resource-efficient AC protocol is possible to be designed using an AEAD scheme. The proposed AC protocol is based on the process (method) presented in Fig. 1. This article contains the following contributions.

1) We propose a Robust AC Protocol for SG-System, called RACP-SG, which employs the AEAD scheme "ASCON" and hash function "ASCON-hash" along with ECC to perform the AC phase. RACP-SG enables the SG system's entities, such as SMs and SEPs, to establish an SK after achieving mutual authentication. Moreover, SMs and SEPs can exchange sensitive information using the established SK. Furthermore, RACP-SG renders the functionality of dynamic SM addition.
2) It is shown through informal security analysis that RACP-SG is resilient against various types of attacks, such as replay, MITM, SM physical capture, and impersonation attacks. It is also shown that RACP-SG ensures untraceability and traceability features. Besides, the SK's security is validated through the well-known random oracle model (ROM). Moreover, the strength of RACP-SG is illustrated through Scyther-based analysis.
3) The performance evaluation shows that RACP-SG incurs lesser computational, communication, and storage overheads, compared to the state of the art, without compromising the security functionalities and features.

B. Paper Organization

The remainder of this article is organized as follows. Related work is presented in Section II. The system model employed for RACP-SG is presented in Section III. Preliminaries are presented in Section IV. The details of the RACP-SG are presented in Section V. The security analysis of RACP-SG is presented in Section VI. The performance evaluation of RACP-SG is presented in Section VII, and the conclusion is presented in Section VIII.

II. Related Work

This section provides an overview of various security schemes proposed for the SG system. Gunduz and Das [3] surveyed various security requirements to ensure secure communication among the SG system entities. Odelu *et al.* [7] proposed an ECC AKE for the SG system. However, their scheme cannot resist MITM, Denial-of-Service (DoS), and ESL attacks and does not provide anonymity and perfect forward secrecy (PFS) features. Li *et al.* [8] proposed a message authentication scheme based on ECC and a secure hash algorithm (SHA-160) for SG systems. However, the scheme is insecure against DoS and impersonation attacks. Similarly, Chen *et al.* proposed a scheme to improve the security of SG systems in [9]. However, as proved in [10], the scheme proposed by Chen *et al.* cannot withstand impersonation and ESL attacks. Kumar *et al.* proposed a scheme in [11] to enhance the security of the SG system. However, their scheme is proved by Yahya *et al.* [12] to be invalid against ESL, SV, and traceability attacks. Bera *et al.* [13] proposed an ECC-based AC protocol for the SG system and utilized ROM to prove the security of the established SK. However, the AC protocol of Bera *et al.* [13] cannot protect De-Synchronization (De-Syn) attacks. Moreover, Bera *et al.* [14] propounded an AC protocol for the Internet of Drones (IoD) to enable secure communication between the drone and ground station. To this end, the scheme uses ECC and SHA-256 to perform the SK establishment process after getting authenticated with the server. However, Chaudhry *et al.* [15] showed that the scheme of Berra *et al.* [13] cannot resist impersonation, MITM, and replay attacks.

Li *et al.* [8] propounded SHA-160 and an ECC-based security scheme for the SG system to ensure indecipherable communication after establishing an SK among the SM and SEPs. However, the scheme cannot resist replay, MITM, and ESL attacks. Besides, the scheme does not ensure SM anonymity and forward secrecy, and cannot provide MA features. An ECC and SHA-160-based AKE scheme was presented by Mahmood *et al.* [16]. However, the scheme cannot resist PI, impersonation, replay, SM capture, and ESL attacks. Besides, the scheme cannot ensure the anonymity of MA and SM. Likewise, Mahmood *et al.* proposed an ECC-based authentication scheme for the SG system in [17], which is again insecure against PI, MITM, replay, impersonation, and SM capture attacks and also does not ensure

TABLE I
SUMMARY OF RELATED AC PROTOCOLS

AC protocol	Year	Primitives utilized	Limitations/Shortcomings
Mahmood *et al.* [16]	2016	SHA-160, XOR, and ECC	Cannot withstand against replay, MITM, impersonation, ESL attacks.
Mahmood *et al.* [17]	2018	SHA-160, XOR, and ECC	Unprotected against DoS, PI, replay, MITM, impersonation, ESL attacks.
Dariush *et al.* [18]	2018	SHA-160, XOR, and ECC	Unprotected against DoS attack. Unable to render SM anonymity and SK security.
Odelu *et al.* [7]	2018	SHA-160, XOR, and ECC	Vulnerable to DoS, MITM, and impersonation attacks. Does not ensure SM anonymity.
Li *et al.* [8]	2019	SHA-160, XOR, and ECC	Cannot restrain replay, MITM, ESL attacks. Unable to provide MA and anonymity features.
Bera *et al.* [13]	2020	SHA-256, XOR, and ECC	Cannot restrain De-Syn attack.
Bera *et al.* [14]	2020	SHA-256, XOR, and ECC	Unsafe against De-Syn attack.
Ayub *et al.* [19]	2020	SHA-160 and XOR	Cannot withstand De-Syn attack.
Tanveer *et al.* [20]	2020	SHA-160, ASCON, and XOR	Cannot restrain De-Syn attack.
Chaudhry *et al.* [21]	2020	SHA-160, ECC, and XOR	Insecure certificate computation. Does not resist device capture attack.
Chaudhry *et al.* [15]	2021	ECC, SHA-160, and XOR	Cannot restrain ESL, DI, SI/SPI, device capture, and SK disclosure attacks.

anonymity and forward secrecy features. Abbasinezhad-Mood and Nikooghadam proposed an ECC-based AKE scheme for SG system in [18]. However, their scheme cannot resist replay attacks and does not ensure SM anonymity. Jo *et al.* presented an ECC-based scheme in [22], which cannot protect ESL and impersonation attacks and does not provide anonymity and untraceability features.

Mahmood *et al.* presented a bi-linear pairing-based authentication scheme in [23] for the SG system. However, the scheme is proved by the authors in [24] to be ineffective against ESL and impersonation attacks. Chaudhry *et al.* proposed a certificate-based AC protocol in [21] for the SG system, which uses ECC and SHA-160. Likewise, Tanveer *et al.* presented an ASCON and SHA-256-based authentication scheme in [20] for the 6LoWPAN environment, which is vulnerable to De-Syn attacks. In the same fashion, the scheme presented by Ayub *et al.* [19] cannot withstand the De-Syn attacks. Wu *et al.* presented a message authentication scheme in [25] for the SG system, which is based on the Diffie–Hellman key exchange mechanism. However, the scheme cannot resist the ESL attack and does not provide anonymity features. Similarly, the scheme presented by Bera *et al.* [13] does not provide the anonymity feature. Badar *et al.* [26] presented an identity-based authentication scheme for SG systems, which uses ECC and a hash function. Likewise, an enhanced pairing-based authentication scheme for the SG system is presented in [27], which uses ECC and hash function. Similarly, Srinivas *et al.* [1] presented a signature-based authentication scheme for SG system, which is able to check various pernicious security attacks. A summary of the related AC protocols is given in Table I.

III. SYSTEM MODEL

A. Authentication Model

To accomplish the AC process in RACP-SG, we consider the authentication model as shown in Fig. 2. There are three components in the model, such that trusted authority (TA), service provider ($SEP_j | j = 1, 2, 3, \ldots, Nse$) where Nse denotes the number of deployed SEP_j, and ($SM_n | n = 1, 2, 3, \ldots, Ns$) where Ns denotes the number of SMs deployed in SG system. The TA is a highly trusted entity and has sufficient computational resources to monitor and control the whole SG system. In RACP-SG, TA is responsible for registering SMs and SEPs, and for system initialization. The SEPs are the organizations that render services to electricity customers and have sufficient computational resources. The SMs are devices with constrained resources, responsible for the electricity consumption and control of the smart home appliances installed in a household. After collecting the sensitive information, SMs transmit the collected information to SEPs via the public Internet. Therefore, to ensure secure information exchange, an AC protocol is imperative in the SG system.

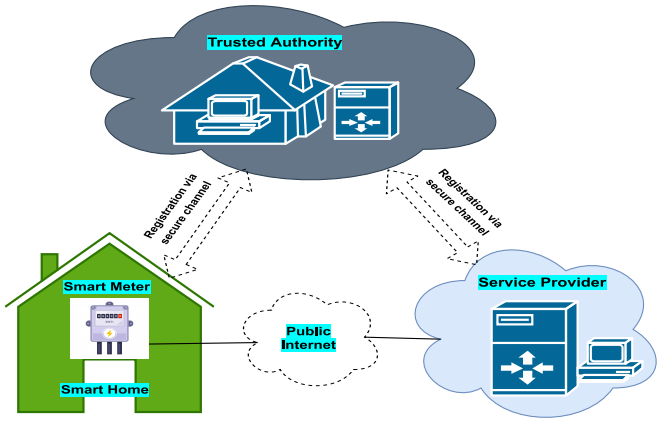

Fig. 2. Application scenario: SG system.

B. Threat Model

We utilize the widely used threat model, i.e., Dolev–Yao (DY) model [28], to validate the strength of RACP-SG against various types of attacks. Moreover, it is assumed that under the DY model, an adversary can potentially access the communicated information since the information is exchanged on public channels. This way, the adversary can capture, delete, or modify the content of the messages being exchanged on the public channel among the communicating nodes. Moreover, the adversary can also capture SM_n physically and can extract, using the power analysis attack, the sensitive information stored in the memory of SM_n. The extracted information can be used to launch various types of attacks, including impersonation and MITM attacks. Furthermore, it is assumed that SM_n is unreliable and untrustworthy while SEP_j is stationed under the physical lock and adversary cannot capture SEP_j physically.

IV. Preliminaries

A. ASCON

ASCON [29] is an online AEAD scheme, which renders confidentiality, authenticity, and integrity of the data simultaneously. The encryption process of ASCON can be defined by the following expression: (CT, APauth) $= \mathcal{E}_K\{(N, \text{AD}), \text{PT}\}$, where CT, APauth, N, AD, K, and PT denote CT, authentication parameter (Tag), nonce, AD, key, and PT, respectively. In addition, the decryption process can be represented by the following expression: (PT, APauth$'$) $= \mathcal{D}_K\{(N, \text{AD}), \text{CT}\}$, where CT, APauth, N, AD, K, and PT denote CT, authentication parameter (Tag), nonce, AD, key, and PT, respectively. The authenticity of the retrieved PT is verified by the equation APauth $=$ APauth$'$. In the proposed RACP-SG, ASCON is used as the encryption/decryption scheme.

B. Physical Unclonable Function

It is assumed that SM_n is equipped with a reliable physical unclonable function (PUF). A PUF function generates the same response to a given input challenge. For two different input challenges, PUF generates different response outputs. PUF takes challenge Ch_{SM_n} as the input and generates response RES, which can be expressed by the expression $\text{RES} = \text{PUF}(\text{Ch}_{\text{SM}_n})$.

C. Fuzzy Extractor

The fuzzy extractor (FE) is utilized to generate a stable secret key. FE comprises two algorithms, such as the key generation algorithm denoted by Gen(·) and the key reproduction algorithm denoted by Rep(·). The Gen(·) generates a stable and unique key Key_{SM_n} and reproduction parameter RP by taking RES as the input, i.e., $\text{Gen(RES)} = (\text{Key}_{\text{SM}_n}, \text{RP})$. The Rep(·) algorithm, which takes RP and RES$'$ as the inputs and reproduces Key_{SM_n}, i.e., $\text{Rep(RES}', \text{RP}) = \text{Key}_{\text{SM}_n}$ provided the condition HMD(RES, RES$'$)$\leq$ ETL, where HMD denotes the Hamming distance and ETL represent the error tolerance. The details of FE can be found in [30] and [31].

V. Proposed RACP-SG Protocol

This section presents an AC protocol, called RACP-SG, for the SG system. The proposed RACP-SG comprises system initialization, SM registration (SMR), AC, Dynamic SM deployment, and phases. Table II tabulates the notations used in the RACP-SG protocol. RACP-SG employs an AEAD scheme known as ASCON, ASCON-hash function, and ECC to design the robust AC protocol for the SG environment. The output size of the ASCON-hash function is 256 bits, we can split it into two chunks to derive a parameter of 128 bits. The ASCON-hash function is faster than SHA-160 and renders the same features as other SHA-160/256. All phases of RACP-SG are described in detail in the succeeding sections.

A. System Initialization Phase

The TA selects an elliptic curve $E_p(m, n)$ over Z_p, where Z_p is the prime field with ensuring condition $4m^3 + 27n^2 \neq 0 \pmod p$. The TA selects a base point P over $E_p(m, n)$ whose order is as big as p, say "N," such that $N \cdot P = O$, where O denotes "zero point." To deploy SEP_j, TA selects an identity ID_{SM_n} and secret key SK_{SEP_j} for a specific SEP_j and computes the public key for SEP_j as $\text{Pbk}_{\text{SEP}_j} = \text{SK}_{\text{SEP}_j} \cdot P$. Finally, TA stores credentials $\{\text{ID}_{\text{SEP}_j}, \text{SK}_{\text{SEP}_j}\}$ in the temper resistance database of SEP_j.

TABLE II
List of Notations Used in RACP-SG

Notation	Description
$PUF(\cdot)$	Physically unclonable function
CH_{SM_n}	Challenge parameter provided as the input to $PUF(\cdot)$
RES_{SM_n}	Response generated by $PUF(\cdot)$
SP, SP_2	Common server parameter generated by SEP_j, which is known only to SEP_j
ID_{SM_n}	Identity of SM_n generated using $PUF(\cdot)$
SID_{SM_n}, PID_{SM_n}	Temporary and real of SM_n
ID_{SEP_j}	Real-Identity and secret key of SEP_j
$E_p(m, n)$, P	Non singular Elliptic Curve with base point P
SK_{SEP_j}, Pbk_{SEP_j}	Private and Public key of SEP_j
$SK_{SEP_j-SM_n}$	Shared secret between SEP_j and SM_n generated using ECC
$SK_{SM_n-SEP_j}$	Shared secret between SM_n and SEP_j generated using ECC
SK_{SM_N}, Pbk_{SM_n}	Private and Public key of SM_n
$(CT_{SEP_j}, APauth_{SEP_j})$	Ciphertext and AP stored at SEP_j
$(CT_x, APauth_x)$	Ciphertext and AP generated by SM_n during AC phase
$(CT_z, APauth_z)$	Ciphertext and AP generated by SEP_j during AC phase
PT_z, PT_{SEP_j}	Plaintext to be encrypted by by using ASCON encryption process at SEP_j
TM_x, TM_z	Timestamps used during RACP-SG's AC phase
T_{DL}, T_{RM}	Allowed delay time and received time of a message
AD_x, AD_z	Associative data used in ASCON's encryption/decryption process
N_x, N_z, N_{SEP_j}	Nonces used in ASCON's encryption/decryption process during AC phase
$\mathcal{E}_K(mg)$, $\mathcal{D}_K(mg)$	ASCON's encryption/decryption of string "mg" using secret key
K_{SM_n}, K_1, Key_{SM_n}	Secret key used in ASCON's encryption/decryption process during AC phase
RN_x, RN_z	Random number used in RACP-SG's AC phase
$Gen(.)$, RP, $Rep(.)$	FE key generation, reproduction parameter, and reproduction function, respectively
$\mathcal{A}$, $H(.)$, $\|$, $\oplus$	Adversary, hash function, concatenation, and XOR, respectively

Definition 1: For any $\text{Pbk}_{\text{SEP}_j} = \text{SK}_{\text{SEP}_j} \cdot P$, $\text{ADV}^{\text{ECDLP}}$ (POT) denotes $\mathcal{A}$'s probability to procure SK_{SEP_j} within polynomial time (POT), which is trivial, and it is also referred to as the elliptic curve discrete logarithm problem (ECDLP).

B. SM Registration Phase

In the SMR phase, SM_x registers itself with TA. TA preloads the secret parameters in the memory of SM_n, which are validated during the AC phase. The following steps are necessary to accomplish the SMR phase.

1) Step SMR-1: SM_n dispatches a registration request message to TA. After receiving the registration request message form SM_n, TA selects random number RN, pseudo identity PID_{SM_n}, and Ch_{SM_n}, each of size 128 bits for an SM_n. In addition, TA computes server parameter SP as $\text{SP} = H(\text{SK}_{\text{SEP}_j} \| \text{ID}_{\text{SEP}_j})$ and temporary identity for SM_n as $\text{SID}_{\text{SM}_n} = (\text{PID}_{\text{SM}_n} \| RN) \oplus SP$. To retrieve the record related to SM_n, TA uses PID_{SM_n} during the AC process. Finally, TA sends the massages $\text{MG}_r : \{\text{Ch}_{\text{SM}_n}, \text{SID}_{\text{SM}_n}\}$ to SM_n via a secure channel.

2) Step SMR-2: After receiving MG_r from TA, SM_n computes $\text{RES}_{\text{SM}_n} = PUF(\text{Ch}_{\text{SM}_n})$, $(\text{Key}_{\text{SM}_n}, \text{RP}) = \text{Gen}(\text{RES}_{\text{SM}_n})$, $Z = H(\text{Key}_{\text{SM}_n})$ and $\text{ID}_{\text{SM}_n} = Z_1 \oplus Z_2$, where Z_1 and Z_2 are two chunks of Z, each of 128 bits. The parameters RES_{SM_n}, Key_{SM_n}, RP, and ID_{SM_n} denote the response, key parameter, reproduction parameter, and real identity of SM_n,

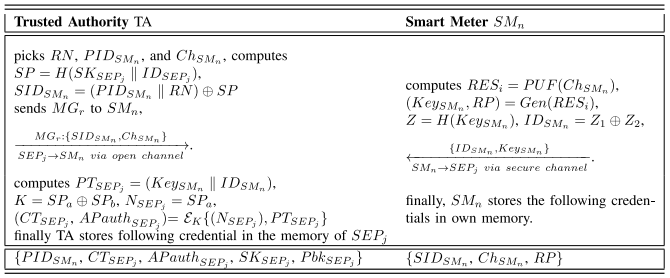

Fig. 3. Registration phase of RACP-SG.

respectively. Finally, SM_n sends $\text{MG}_{r2} : \{\text{Key}_{\text{SM}_n}, \text{ID}_{\text{SM}_n}\}$ to TA via a secure channel and stores the credentials $\{\text{SID}_{\text{SM}_n}, \text{Ch}_{\text{SM}_n}, \text{RP}\}$ in its own memory.

3) Step SMR-3: After receiving MG_{r2}, it computes $\text{PT}_{\text{SEP}_j} = (\text{Key}_{\text{SM}_n} \| \text{ID}_{\text{SM}_n})$, where PT_{SEP_j} is the PT. In addition, TA computes $K = \text{SP}_a \oplus \text{SP}_b$, where SP_a and SP_b are two chunks of SP, each of 128 bits, and nonce $N_{\text{SEP}_j} = \text{SP}_a$. Moreover, TA computes $(\text{CT}_{\text{SEP}_j}, \text{APauth}_{\text{SEP}_j}) = \mathcal{E}_K\{(N_{\text{SEP}_j}), \text{PT}_{\text{SEP}_j}\}$ by using the ASCON encryption process. Finally, TA stores the credentials $\{\text{PID}_{\text{SM}_n}, \text{CT}_{\text{SEP}_j}, \text{APauth}_{\text{SEP}_j}, \text{SK}_{\text{SEP}_j}, \text{Pbk}_{\text{SEP}_j}\}$ in the memory SEP_j. The SMR phase is summarized in Fig. 3.

C. AC Phase

In this phase, SM_n achieves the authentication with SEP_j and establishes an SK with SEP_j for indecipherable communication in the future. To establish an SK, both SM_n and SEP_j require to execute the following steps.

1) Step AC-1: SM_n selects TM_x, SK_{SM_n}, and RN_x of size 32, 160, and 128 bits, respectively. Moreover, SM_n computes

$$\text{Pbk}_{\text{SM}_n} = \text{SK}_{\text{SM}_n} \cdot P \tag{1}$$

$$\text{SK}_{\text{SEP}_j - \text{SM}_n} = \text{SK}_{\text{SM}_n} \cdot \text{Pbk}_{\text{SEP}_j} \tag{2}$$

$$Y = H(\text{SID}_{\text{SM}_n} \| \text{SK}_{\text{SM}_n - \text{SEP}_j} \| \text{TM}_x \| \text{Pbk}_{\text{SM}_n}) \tag{3}$$

where $\text{SK}_{\text{SEP}_j - \text{SM}_n}$ denotes the shared secret key and $\text{Pbk}_{\text{SEP}_j}$ denotes the public key of SEP_j. In addition, SM_n splits Y equally into two parts, Y_a and Y_b, each with 128 bits. Moreover, SM_n computes the secret key as $K_{\text{SM}_n} = Y_a \oplus Y_b$, which is used in the encryption process and AD as $N_x = Y_a$. Furthermore, SM_n by using the ASCON encryption algorithm computes

$$(\text{CT}_x, \text{APauth}_x) = \mathcal{E}_{K_{\text{SM}_n}}\{(N_x), \text{RN}_x\} \tag{4}$$

where CT_x and APauth_x are the CT and authentication parameter. Finally, SM_n fabricates the message $\text{MG}_1 : \{\text{TM}_x, \text{SID}_{\text{SM}_n}, \text{CT}_x, \text{APauth}_x, \text{Pbk}_{\text{SM}_n}\}$ and dispatches MG_1 to SEP_j via open channel.

2) Step AC-2: Upon procuring MG_1 from SM_n, the freshness of the received MG_1 is checked by SEP_j by validating the condition $T_{\text{DL}} \geq |T_{\text{RM}} - \text{TM}_x|$, where T_{DL} and TM_x denote the allowed time delay and generation time of MG_1, respectively. SEP_j terminates the AC process if SEP_j fails to validate the condition. Otherwise, SEP_j computes

$$\text{SK}_{\text{SEP}_j - \text{SM}_n} = \text{SK}_{\text{SEP}_j} \cdot \text{Pbk}_{\text{SM}_n} \tag{5}$$

$$Z = H(\text{SID}_{\text{SM}_n} \| \text{SK}_{\text{SEP}_j - \text{SM}_n} \| \text{TM}_x \| \text{Pbk}_{\text{SM}_n}) \tag{6}$$

where $\text{SK}_{\text{SEP}_j - \text{SM}_n}$ denotes the shared secret key and Pbk_{SM_n} represents the public key of SM_n. In addition to this, SEP_j splits Z into Z_a and Z_b, each of 128 bits. Additionally, to accomplish the decryption process, SEP_j determines the secret key as $K_{\text{SEP}_j} = Z_a \oplus Z_b$ and AD as $N_y = Z_a$. Furthermore, SEP_j by using ASCON decryption algorithm computes

$$(\text{PT}_y, \text{APauth}_y) = \mathcal{D}_{K_{\text{SEP}_j}}\{(N_y), \text{CT}_x\}. \tag{7}$$

Finally, the condition $\text{APauth}_x = \text{APauth}_y$ is validated by SEP_j to check the authenticity of the received MG_1. If it holds, SEP_j contemplates MG_1 as a valid message and extracts $\text{PT}_y = \{\text{RN}_x\}$ from the decryption process of ASCON. Otherwise, SEP_j terminates the AC process.

3) Step AC-3: After validating the authenticity of MG_1, SEP_j computes $\text{SP}_2 = H(\text{SK}_{\text{SEP}_j} \| \text{ID}_{\text{SEP}_j})$ and extracts PID_{SM_n} and RN_{SM_n} from the received SID_{SM_n} as follows:

$$(\text{PID}_{\text{SM}_n} \| \text{RN}_{\text{SM}_n}) = \text{SID}_{\text{SM}_n} \oplus \text{SP}_2. \tag{8}$$

In addition to this, SEP_j checks if PID_{SM_n} exists in its own database. If it is found, SEP_j retrieves stored information $\{\text{CT}_{\text{SEP}_j}, \text{APauth}_{\text{SEP}_j}\}$ related to PID_{SM_n}. Moreover, SEP_j determines the secret key as $K_1 = \text{SP}_2^a \oplus \text{SP}_2^b$, where SP_2^a and SP_2^b are derived by dividing SP into two equal parts and AD $N_{\text{SEP}_j} = \text{SP}_2^a$. Furthermore, SEP_j extracts ID_{SM_n} and Key_{SM_n} associated with SM_n as follows:

$$((\text{ID}_{\text{SM}_n} \| \text{Key}_{\text{SM}_n}), \text{APauth}'_{\text{SEP}_j}) = \mathcal{D}_{K_1}\{(N_{\text{SEP}_j}), \text{CT}_{\text{SEP}_j}\}. \tag{9}$$

To validate the authenticity of the data stored at SEP_j, SEP_j requires to check the condition $\text{APauth}'_{\text{SEP}_j} = \text{APauth}_{\text{SEP}_j}$. If it holds, SEP_j continues the AC process. Otherwise, SEP_j terminates the AC process.

4) Step AC-4: After extracting ID_{SM_n} and Key_{SM_n}, SEP_j picks TM_z, RN_z, and $\text{RN}^n_{\text{SM}_n}$ with a size of 32, 128, and 128 bits, respectively. Moreover, SEP_j computes

$$(\text{PID}_{\text{SM}_n} \| \text{RN}^n_{\text{SM}_n}) \oplus \text{SP}_2 = \text{SID}^{\text{new}}_{\text{SM}_n} \tag{10}$$

where $\text{SID}^{\text{new}}_{\text{SM}_n}$ is a new temporary identity, which will be used by SM_n to achieve anonymous communication. Moreover, SEP_j calculates

$$Q = H(\text{SK}_{\text{SEP}_j - \text{SM}_n} \| \text{ID}_{\text{SM}_n} \| \text{Key}_{\text{SM}_n} \| \text{TM}_z \| \text{RN}_x) \tag{11}$$

and determines $\text{AD}_z = Q_a \oplus Q_b$, $N_z = Q_a$, and $\text{PT}_z = (\text{SID}^{\text{new}}_{\text{SM}_n} \| \text{RN}_z)$, where AD_z, N_z, and PT_z are the AD, nonce, and PT, respectively. Furthermore, SEP_j by using the ASCON encryption process computes $(\text{CT}_z, \text{APauth}_z) = \mathcal{E}_{\text{Key}_{\text{SM}_n}}\{(N_z, \text{AD}_z), \text{PT}_z\}$. In addition, SEP_j computes the SK by computing $\text{SK}_{\text{SEP}_j} = H(Q \| \text{RN}_x \| \text{RN}_z \| \text{TM}_z)$ to achieve the encrypted communication in future. Finally, SEP_j composes a message $\text{MG}_2 : \{\text{TM}_z, \text{CT}_z, \text{APauth}_z\}$ and dispatches it to SM_n via an open channel.

5) Step AC-5: After receiving MG_2 from SEP_j, SM_n validates the freshness of the received message by verifying the condition $T_{\text{DL}} \geq |T_{\text{RM}} - \text{TM}_z|$, where T_{DL} and TM_z denote the allowed time delay and generation time of MG_2, respectively. After the condition is successfully validated, SM_n computes $\text{RES}_{\text{SM}_n} = \text{PUF}(\text{Ch}_{\text{SM}_n})$ and $\text{Key}_{\text{SM}_n} = \text{Rep}(\text{RES}_i, \text{RP})$, where Ch_{SM_n}, RES_{SM_n}, and Key_{SM_n} are the

Smart Meter SM_n	**Service Provider** SEP_j
$\{SID_{SM_n}, Ch_{SM_n}, RP\}$	$\{PID_{SM_n}, CT_{SEP_j}, APauth_{SEP_j}, SK_{SEP_j}, Pbk_{SEP_j}\}$
picks TM_x, SK_{SM_n}, RN_x, and computes, $Pbk_{SM_n} = SK_{SM_n} \cdot P$, $SK_{SEP_j-SM_n} = SK_{SM_n} \cdot Pbk_{SEP_j}$, $Y = H(SID_{SM_n} \parallel SK_{SM_n-SEP_j} \parallel TM_x \parallel Pbk_{SM_n})$, splits Y into two chunks Y_a and Y_b, $K_{SM_n} = Y_a \oplus Y_b$, $N_x = Y_a$, $(CT_x, APauth_x)$=$\mathcal{E}_{K_{SM_n}}$ $\{(N_x), RN_x\}$, $\xrightarrow[SM_n \rightarrow SEP_j]{\{TM_x,\ SID_{SM_n},\ CT_x,\ APauth_x,\ Pbk_{SM_n}\}}$.	checks $T_{DL} \geq \lvert T_{RM} - TM_x \rvert$, if holds, computes $SK_{SEP_j-SM_n} = SK_{SEP_j} \cdot Pbk_{SM_n}$, $Z = H(SID_{SM_n} \parallel SK_{SEP_j-SM_n} \parallel TM_x \parallel Pbk_{SM_n})$, splits Z into two chunks Z_a and Z_b, $K_{SEP_j} = Z_a \oplus Z_b$, $N_y = Z_a$, $(PT_y, APauth_y)$= $\mathcal{D}_{K_{SEP_j}}$ $\{(N_y), CT_x\}$, validates $APauth_x = APauth_y$, if so, extracts $PT_y = \{RN_x\}$ from decryption process. computes $SP_2 = H(SK_{SEP_j} \parallel ID_{SEP_j})$, extracts $(PID_{SM_n} \parallel RN_{SM_n}) = SID_{SM_n} \oplus SP_2$, checks if PID_{SM_n} exists, if so, retrieves $\{CT_{SEP_j}, APauth_{SEP_j}\}$, computes $K_1 = SP_2^a \oplus SP_2^b$, $N_{SEP_j} = SP_2^a$, $((ID_{SM_n} \parallel Key_{SM_n}), APauth'_{SEP_j}) = \mathcal{D}_{K_1}$ $\{(N_{SEP_j}), CT_{SEP_j}\}$, checks $APauth'_{SEP_j} = APauth_{SEP_j}$, if holds, picks TM_z, RN_z, $RN^n_{SM_n}$, computes $(PID_{SM_n} \parallel RN^n_{SM_n}) \oplus SP_2 = SID^{new}_{SM_n}$, Q=$H(SK_{SEP_j-SM_n} \parallel ID_{SM_n} \parallel Key_{SM_n} \parallel TM_z \parallel RN_x)$, $AD_z = Q_a \oplus Q_b$, $N_z = Q_a$, $PT_z = (SID^{new}_{SM_n} \parallel RN_z)$, $SK_{SEP_j} = H(Q \parallel RN_x \parallel RN_z \parallel TM_z)$ $(CT_z, APauth_z) = \mathcal{E}_{Key_{SM_n}}$ $\{(N_z, AD_z), PT_z\}$, $\xleftarrow[SEP_j \rightarrow SM_n]{\{TM_z,\ CT_z,\ APauth_z\}}$.
checks $T_{DL} \geq \lvert T_{RM} - TM_z \rvert$, if holds, computes $RES_i = PUF(Ch_{SM_n})$, $Key_{SM_n} = Rep(RES_i, RP)$, $ZZ = H(Key_{SM_n})$, $ID_{SM_n} = ZZ_1 \oplus ZZ_2$, G=$H(SK_{SM_n-SEP_j} \parallel ID_{SM_n} \parallel Key_{SM_n} \parallel TM_z \parallel RN_x)$, $AD_w = G_a \oplus G_b$, $N_w = G_a$, $(PT_z, APauth_w) = \mathcal{D}_{Key_{SM_n}}$ $\{(N_w, AD_w), CT_z\}$, checks condition $APauth_z = APauth_w$, if holds, retrieves $PT_z = (SID^{new}_{SM_n} \parallel RN_z)$, updates SID_{SM_n} with $SID^{new}_{SM_n}$, computes $SK_{SM_n} = H(G \parallel RN_x \parallel RN_z \parallel TM_z)$.	
$SK_{SM_n} (= SK_{SEP_j}) = H(H((SK_{SM_n} \cdot SK_{SEP_j} \cdot P) \parallel ID_{SM_n} \parallel Key_{SM_n} \parallel TM_z) \parallel RN_x \parallel RN_z \parallel TM_z)$	

Fig. 4. RACP-SG AC phase.

challenge, response, and generated key, respectively. In addition, SM_n computes the identity as $ZZ = H(\text{Key}_{SM_n})$ and $\text{ID}_{SM_n} = ZZ_1 \oplus ZZ_2$. Moreover, SM_n computes

$$G = H(\text{SK}_{SM_n-SEP_j} \| \text{ID}_{SM_n} \| \text{Key}_{SM_n} \| \text{TM}_z \| \text{RN}_x) \quad (12)$$

and splits G into G_a and G_b, each of 128 bits. In addition to this, SM_n determines $\text{AD}_w = G_a \oplus G_b$ and $N_w = G_a$. Here, AD_w and N_w denote AD and nonce, respectively. To determine the PT, SM_n by using ASCON computes

$$(\text{PT}_z, \text{APauth}_w) = \mathcal{D}_{\text{Key}_{SM_n}}\{(N_w, \text{AD}_w), \text{CT}_z\}. \quad (13)$$

Moreover, SM_n checks the condition $\text{APauth}_z = \text{APauth}_w$ to validate the authenticity of the received message. If it holds, SM_n retrieves $\text{PT}_z = (\text{SID}^{\text{new}}_{SM_n} \| \text{RN}_z)$ from the decryption process of ASCON. Furthermore, SM_n updates SID_{SM_n} with $\text{SID}^{\text{new}}_{SM_n}$ to achieve the anonymous communication in the future. Finally, SM_n computes the SK as $\text{SK}_{SM_n} = H(G \| \text{RN}_x \| \text{RN}_z \| \text{TM}_z)$ to ensure indecipherable communication in future. The AC phase is summarized in Fig. 4.

D. Dynamic SM Addition Phase

In this phase, TA deploys a new SM_n by performing the same procedure as described in Section V-B from Step SMR-1 to Step SMR-3. However, to deploy a new SM_n, TA needs to select new parameters, such as RN^{new}, $PID^{\text{new}}_{SM_n}$, and $Ch^{\text{new}}_{SM_n}$.

VI. Security Analysis

The security analysis of the proposed RACP-SG is presented in this section. First, an informal analysis is presented to show that RACP-SG is secure against various security attacks, including MITM, impersonation, and replay attacks. Then, SK security is establish through the well-known ROM. Finally, the Scyther tool is used to show that RACP-SG is secure against various covert attacks.

A. Informal Security Analysis

This section proffers the informal security analysis of RACP-SG to illustrates RACP-SG's resiliency to resist various pernicious security attacks, such as MITM, impersonation, replay, and SM capture attacks.

1) Untraceability and Anonymity: There are two messages, such that MG_1 : $\{\text{TM}_x, \text{SID}_{SM_n}, \text{CT}_x, \text{APauth}_x, \text{Pbk}_{SM_n}\}$ and MG_2 : $\{\text{TM}_z, \text{CT}_z, \text{APauth}_z\}$, which are exchanged to accomplish the AC process in RACP-SG. SID_{SM_n} is computed as $\text{SID}_{SM_n} = (\text{PID}_{SM_n} \| \text{RN}) \oplus SP$, where PID_{SM_n} is the identity, which is used to search the record related to SM_n. PID_{SM_n} cannot be derived from SID_{SM_n} because it is protected by $\text{SP} = H(\text{ID}_{SEP_j} \| K_{SEP_j})$, where ID_{SEP_j} and K_{SEP_j} are known only to SEP_j. Therefore, $\mathcal{A}$ cannot extract PID_{SM_n} from SID_{SM_n}. In addition, SEP_j generates $\text{SID}^{\text{new}}_{SM_n}$, by selecting a fresh random number, and sends it to SM_n in the encrypted form. This new $\text{SID}^{\text{new}}_{SM_n}$ is used by SM_n during the new AC session. Therefore, it is hard for $\mathcal{A}$ to extract any parameter from seized messages that could enable it to trace SM_n and SEP_j. This suggests that RACP-SG provides the anonymity feature. Furthermore, MG_1 and MG_2 change dynamically and randomly for each new AC session in RACP-SG, making it impossible for $\mathcal{A}$ to relate the captured messages, such as MG_1 and MG_2 from two different AC sessions to extract any useful information. This suggests that RACP-SG ensures the untraceability feature.

2) De-Syn Attack: The De-Syn attack is possible only when the entities involved in the AC process update some of the parameters during the execution of each AC process to ensure anonymous communication. In RACP-SG, SID_{SM_n} is updated by SEP_j during the execution of every new AC session. SID_{SM_n} is computed as $\text{SID}_{SM_n} = (\text{PID}_{SM_n} \| \text{RN}) \oplus \text{SP}$, where PID_{SM_n} represents the smart meter SM_n and RN is the random number. SEP_j constructs a new $\text{SID}^{\text{new}}_{SM_n} = (\text{PID}_{SM_n} \| \text{RN}^n_{SM_n}) \oplus \text{SP}_2$, here PID_{SM_n} remains the same while SEP_j selects new random number $\text{RN}^n_{SM_n}$ generate new $\text{SID}^{\text{new}}_{SM_n}$. SM_n uses

this $\text{SID}_{\text{SM}_n}^{\text{new}}$ during each new AC session. $\mathcal{A}$ cannot effectuate a De-Syn attack by drooping any of the message communicated during the AC process because PID_{SM_n} remains the same, which is used to retrieve the record related to SM_n. This suggests that RACP-SG is able to resist the De-Syn attack.

3) MITM Attack: According to the threat model defined in Section III-B, $\mathcal{A}$ can expropriate all the communicated messages, such as $\text{MG}_1 : \{\text{TM}_x, \text{SID}_{\text{SM}_n}, \text{CT}_x, \text{APauth}_x, \text{Pbk}_{\text{SM}_n}\}$ and $\text{MG}_2 : \{\text{TM}_z, \text{CT}_z, \text{APauth}_z\}$ that are exchanged during the AC process. After capturing MG_1, $\mathcal{A}$ can generate a bogus message MG_1' to make SEP_j believe that MG_1' is from a legitimate SM_n. However, without knowing the secret credentials $\{\text{SK}_{\text{SM}_n}, \text{SK}_{\text{SM}_n-\text{SEP}_j}, \text{PID}_{\text{SM}_n}\}$, it is impractical for $\mathcal{A}$ to fabricate a valid MG_1. Similarly, it is hard for $\mathcal{A}$ to generate MG_2 without knowing the secret credentials, i.e., $\{\text{SK}_{\text{SEP}_j}, \text{SK}_{\text{SEP}_j-\text{SM}_n}, \text{ID}_{\text{SM}_n}, \text{Key}_{\text{SM}_n}\}$. This suggests that RACP-SG is able to resist the MITM attack.

4) Replay Attack: According to the threat model defined in Section III-B, $\mathcal{A}$ can expropriate all the exchanged messages, i.e., $\text{MG}_1 : \{\text{TM}_x, \text{SID}_{\text{SM}_n}, \text{CT}_x, \text{APauth}_x, \text{Pbk}_{\text{SM}_n}\}$ and $\text{MG}_2 : \{\text{TM}_z, \text{CT}_z, \text{APauth}_z\}$, and can attempt to replay the captured message to entities of the SG system to obtain valuable information from the entities. In RACP-SG, all the entities of the SG system, which are involved in the AC process, are time synchronized. In addition, all the communicated messages, i.e., MG_1 and MG_1, incorporate the latest timestamp and fresh random number. A message receiving entity ensures the freshness of the received message by checking the condition $T_{\text{DL}} \geq |T_{\text{RM}} - \text{TM}_x|$ and $T_{\text{DL}} \geq |T_{\text{RM}} - \text{TM}_z|$ for MG_1 and MG_2, respectively. On failure, the corresponding entity drops the received message and terminates the AC process. In this way, the message receiving entity can detect the replay attack. Therefore, RACP-SG is able to resist the replay attack.

5) SM Capture Attack: According to the threat model defined in Section III-B, $\mathcal{A}$ can capture an SM_n deployed in the SG system and can extract secret parameters, such as $\{\text{SID}_{\text{SM}_n}$ and $\text{Ch}_{\text{SM}_n}, \text{RP}\}$ stored in the memory of the SM_n by employing the power analysis attack. However, from the secret credentials obtained from a captured SM_n, $\mathcal{A}$ cannot procure secret credentials of other noncaptured or noncompromised SM_n. In addition, the secret credentials are different for every deployed SM_n. Therefore, $\mathcal{A}$ cannot breach the security of the noncompromised SM_n. Hence, the proposed RACP-SG is able to resist the SM_n capture attack.

6) Impersonation Attack: There are two messages, such as $\text{MG}_1 : \{\text{TM}_x, \text{SID}_{\text{SM}_n}, \text{CT}_x, \text{APauth}_x, \text{Pbk}_{\text{SM}_n}\}$ and $\text{MG}_2 : \{\text{TM}_z, \text{CT}_z, \text{APauth}_z\}$, which are communicated to accomplish the AC process in RACP-SG. $\mathcal{A}$ can impersonate as legitimate SM_n by generating a valid MG_1. However, $\mathcal{A}$ cannot generate a licit message MG_1 on behalf of SM_n without knowing the secret parameters, such as $\{\text{SK}_{\text{SM}_n}, \text{SK}_{\text{SM}_n-\text{SEP}_j}, \text{PID}_{\text{SM}_n}\}$. Therefore, $\mathcal{A}$ cannot effectuate the SM impersonation attack. Similarly, $\mathcal{A}$ cannot impersonate as SEP_j without knowing the secret credentials, such as $\{\text{SK}_{\text{SEP}_j}, \text{SK}_{\text{SEP}_j-\text{SM}_n}, \text{ID}_{\text{SM}_n}, \text{Key}_{\text{SM}_n}\}$. Hence, the proposed RACP-SG can resist the SEP/SM impersonation attack.

TABLE III
DESCRIPTION OF DIFFERENT ROM QUERIES

Query	Purpose
$Send(\Pi^{p1}, Msg)$	This query is initiated by $\mathcal{A}$ to launch an active attack by iteratively transmitting the message Msg to Π^{p1} with Π^{p1} also supposed to be responding.
$Test(\Pi^{p1})$	This query is leveraged by $\mathcal{A}$ to verify the validity of SK, i.e., if or not SK is a real or random output of a flipped coin 'B'.
$Reveal(\Pi^{p1})$	Making this query enables $\mathcal{A}$ to obtain the SK that is established between Π^{p1} and its partner entity.
$CorruptSM(\Pi_{SM_n}^{p2})$	This query leads $\mathcal{A}$ to, by means of power analysis attack, extract the sensitive information from memory of SM_n.
$Execute(\Pi_{SM_n}^{p2}, \Pi_{SEP_j}^{p3})$	$\mathcal{A}$ can leverage this query to access all the messages transmitted between SM_n and SEP_j.

7) ESL Attack: In RACP-SG, the SK is computed as $\text{SK}_{\text{SM}_n} = (\text{SK}_{\text{SEP}_j}) = H(H((\text{SK}_{\text{SM}_n} \cdot \text{SK}_{\text{SEP}_j} \cdot P) \| \text{ID}_{\text{SM}_n} \| \text{Key}_{\text{SM}_n} \| \text{TM}_z) \| \text{RN}_x \| \text{RN}_z \| \text{TM}_z)$, which is the amalgamation of long term secret (LTS)$\{\text{ID}_{\text{SM}_n}, \text{Key}_{\text{SM}_n}\}$ and ephemeral secrets (ES), such as $\{\text{RN}_x, \text{RN}_Z, \text{SK}_{\text{SM}_n}, \text{SK}_{\text{SEP}_j}\}$. Therefore, to break the security of the establish SK, $\mathcal{A}$ requires to know both the LTS and ES. Thus, RACP-SG is capable of withstanding the ESL attack.

B. ROM-Based Formal Security Analysis

In this section, we present the ROM-based analysis of our proposed RACP-SG protocol in order to verify security of the SK that is established between SM_n and SEP_j. Let Π_{RC}^{p1}, $\Pi_{\text{SM}_n}^{p2}$, and $\Pi_{\text{SEP}_j}^{p3}$ denote instances $p1$, $p2$, and $p3$ of the participants RA, SM_n, and SEP_j, also called oracles. ROM has various components, as detailed in [32], which are associated with the various queries used by $\mathcal{A}$.

The DY model designates that $\mathcal{A}$ can expropriate all the messages propagated between the entities in the SG environment. This signifies that $\mathcal{A}$, utilizing the queries represented in Table III, can modify, inject, and delete the communicated messages. Moreover, this also designates that $\mathcal{A}$ can access the hash function $H(\cdot)$, which is represented as a random-oracle, say ASHsh. Above this, the queries, represented in Table III, are utilized by $\mathcal{A}$ to simulate an attack.

SK's Semantic Security: $\mathcal{A}$ needs to differentiate an instance's real SK from a random number, under ROM. Furthermore, $\mathcal{A}$ has the ability to perform many Test queries to either Π^{p1} or Π^{p2}. At the end of the game, bit B' is guessed by $\mathcal{A}$. $\mathcal{A}$ will win the game if $B = B'$. $\mathcal{A}$'s advantage in breaching SK's security semantics is denoted by $\text{ADV}_{\mathcal{A}}^{\text{RACP-SG}}(\text{POT}) = |2 \cdot Prb[SU] - 1|$, where SU represents the event, in which $\mathcal{A}$ can win the game. RACP-SG is secure if $\text{ADV}_{\mathcal{A}}^{\text{RACP-SG}}(\text{POT})$ is insignificant under ROM.

Definition 2: Let polynomial-time $\mathcal{A}$ execute against the AEAD scheme and perform at maximum QU queries of range space LN to the encryption/decryption oracle. Then, the online chosen CT attack (OCCA3) advantage of $\mathcal{A}$ can be defined as [32]–[34]

$$\text{ADV}_{\varphi}^{\text{OCCA3}}(\mathcal{A}) \leq \text{ADV}_{\varphi}^{\text{OPRP-CPA}}(\text{QU}, \text{LN}, \text{POT}) + \text{ADV}_{\varphi}^{\text{INT-CTXT}}(\text{QU}, \text{LN}, \text{POT}) \quad (14)$$

where ADV, INT–CTXT, and OPRP–CPA denote the advantage, integrity of CT, and online pseudorandom permutation chosen-plaintext attack, respectively.

Theorem 1: Let $\mathcal{A}$ running against RACP-SG in POT to obtain the SK, which is established between smart meter SM_n and service provider SEP_j. If H_{qu} signifies hash function (ASCON-hash) quires, |ASHsh| denotes range space of hash function (ASCON-hash) output, H_{pf}^2 represents the PUF queries, |PUF| denotes the range space of PUF. $ADV_{\mathcal{A}}^{ECDLP}(POT)$ and $ADV_{ASCON,\mathcal{A}}^{OCCA3}(QU, LN, POT)$ represent the advantage of $\mathcal{A}$ in solving ECDLP and breaking the security ASCON (Definition 2), respectively. The advantage of $\mathcal{A}$ in breaking RACP-SG's security, for procuring the SK, established between SM_n and SEP_j can be represented as

$$ADV_{\mathcal{A}}^{RACP-SG}(POT) \leq \frac{H_{qu}^2}{|ASHsh|} + \frac{H_{pf}^2}{|PUF|} + 2.ADV_{ASCON,\mathcal{A}}^{OCCA3}(QU, LN, POT) + ADV_{\mathcal{A}}^{ECDLP}(POT). \quad (15)$$

Proof: The security of established SK is validated in the succeeding five games ($Gm_x|x = 0, 1, 2, 3, 4$) by employing the queries presented in Table III.

Gm_0: A real attack is launched by $\mathcal{A}$ on RACP-SG under ROM. It is necessary for $\mathcal{A}$ to imagine the bit b at the start of Gm_0. Then, we get

$$ADV_{\mathcal{A}}^{RACP-SG}(POT) = |2.Prb[SU0] - 1|. \quad (16)$$

Gm_1: An eavesdrop attack is effectuated by $\mathcal{A}$ in this game. $\mathcal{A}$ by using Execute($\Pi_{SM_n}^{p2}, \Pi_{SEP_j}^{p3}$) query captures all the messages, such as $MG_1 : \{TM_x, SID_{SM_n}, CT_x, APauth_x, Pbk_{SM_n}\}$ and $MG_2 : \{TM_z, CT_z, APauth_z\}$, which are exchanged during the AC process. After capturing the messages, $\mathcal{A}$ attempts to construct the SK, which is computed as $SK_{SM_n}(= SK_{SEP_j}) = H(H((SK_{SM_n} \cdot SK_{SEP_j} \cdot P)\|ID_{SM_n}\|Key_{SM_n}\|TM_z)\|RN_x\|RN_z\|TM_z)$ by making the Test and Reveal queries. AS SK is constructed by using both LTS and ES credentials. Therefore, to derive SK, $\mathcal{A}$ needs to know both ES and LTS credentials. The probability of deriving SK, only by capturing the exchanged messages during AC process, will not increase at all. Now, under eavesdrop attack both Gm_0 and Gm_1 become indistinguishable. Therefore, we have

$$Prb[SU0] = Prb[SU1]. \quad (17)$$

Gm_2: Gm_2 effectuates an active attack by simulating *Send* and ASHsh queries. In Gm_2, $\mathcal{A}$ requires to make believe an entity into receiving a bogus (modified) message. $\mathcal{A}$ is allowed to make several ASHsh queries to find the collisions in hash digests. Since, all the communicated messages MG_1 and MG_2 indirectly incorporate entity's identity, SP/SP_2, and LTS, which are protected by ASCON-hash. $\mathcal{A}$ makes several ASHsh/Send queries to find the collision. However, it is hard for $\mathcal{A}$ to find collision because the ASCON-hash function is collision resistant. Therefore, by the birthday paradox

$$|Prb[SU1] - Prb[SU2]| \leq \frac{H_{qu}^2}{2|ASHsh|}. \quad (18)$$

Gm_3: In this game, $\mathcal{A}$ launches an active attack by simulating CorruptSM($\Pi_{SM_n}^{p2}$) query. For this purpose, $\mathcal{A}$ by capturing one or more SM_n extract all the sensitive information stored in its memory using the power analysis attack. $\mathcal{A}$ cannot compute the SKs established between SEP_j and other noncompromised SM_n in the SG system as the parameters challenge Ch_{SM_n} is distinct for each SM_n. In addition, PUF generates unique output (response) for every input (challenge). Due this property of PUF function, it is infeasible for $\mathcal{A}$ to find the collision (same output for two different inputs). According to Gm_3, it follows:

$$|Prb[SU3] - Prb[SU2]| \leq \frac{H_{pf}^2}{2|PUF|}. \quad (19)$$

Gm_4: In this game, $\mathcal{A}$ captures all the messages, such as MG_1:{TM_x, SID_{SM_n}, CT_x, $APauth_x$, Pbk_{SM_n}} and $MG_2 : \{TM_z, CT_z, APauth_z\}$, which are exchanged during the AC process. $\mathcal{A}$ attempts to decrypt CT_x and CT_z to procure the sensitive information required to construct the SK. As the CT_x is encrypted with the OCCA3 secure AEAD scheme, known as ASCON (Definition 2) and using the key $K_{SM_n} = Y_a \oplus Y_b$, which is derived from $Y = H(SID_{SM_n}\|SK_{SM_n-SEP_j}\|TM_x\|Pbk_{SM_n})$. $\mathcal{A}$ cannot derive the parameter Y because it contains $SK_{SM_n-SEP_j}$, which is derived as $SK_{SEP_j-SM_n} = SK_{SM_n} \cdot Pbk_{SEP_j}$. In addition, it is hard for $\mathcal{A}$ to derive the secret key SK_{SM_n} from public key Pbk_{SM_n} of SM_n because to derive SK_{SM_n} from Pbk_{SM_n} is an ECDLP as demonstrated in (Definition 1). This concludes that

$$|Prb[SU3] - Prb[SU4]| \leq ADV_{ASCON,\mathcal{A}}^{OCCA3}(QU, LN, POT) + ADV_{\mathcal{A}}^{ECDLP}(POT). \quad (20)$$

After executing all queries, $\mathcal{A}$ needs to presume bit B' for winning the game after making the Test query. It is then obvious that

$$Prb[SU4] = 1/2. \quad (21)$$

From (16) and (17), we get

$$ADV_{\mathcal{A}}^{RACP-SG}(POT) = |2.Prb[SU0] - \frac{1}{2}|. \quad (22)$$

From (22), we get

$$\frac{1}{2}.ADV_{\mathcal{A}}^{RACP-SG}(POT) = |Prb[SU0] - \frac{1}{2}|. \quad (23)$$

By using (21) and (23), we obtain

$$\frac{1}{2}.ADV_{\mathcal{A}}^{RACP-SG}(POT) = |Prb[SU1] - Prb[SU4]|. \quad (24)$$

By using the triangular inequality, we get

$$\begin{aligned} |Prb[SU1] - Prb[SU4]| &\leq |Prb[SU1] - Prb[SU2]| \\ &+ |Prb[SU2] - Prb[SU4]| \\ &\leq |Prb[SU1] - Prb[SU2]| + |Prb[SU2] - Prb[SU3]| \\ &+ |Prb[SU3] - Prb[SU4]|. \end{aligned} \quad (25)$$

By using (18)–(20) and (25), we get

$$ADV_{\mathcal{A}}^{RACP-SG}(POT) \leq \frac{H_{qu}^2}{|ASHsh|} + \frac{H_{pf}^2}{|PUF|} + 2.ADV_{ASCON,\mathcal{A}}^{OCCA3}(QU, LN, POT) + ADV_{\mathcal{A}}^{ECDLP}(POT). \quad (26)$$

■

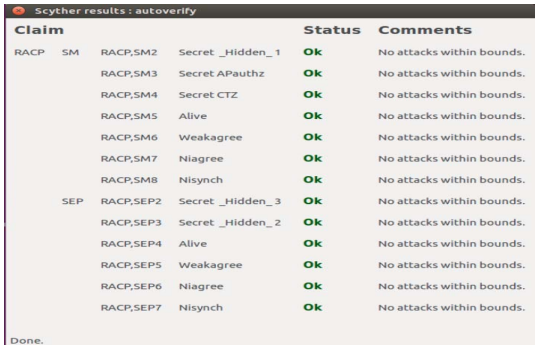

Fig. 5. Scyther-based Security Analysis of RACP-SG.

TABLE IV
COMPUTATIONAL TIME OF VARIOUS CRYPTOGRAPHIC PRIMITIVES

Cryptographic Primitive	Notations	Computational Time (Raspberry Pi-3)	Server (SEP_j)
ECC point Multiplication	T_{ECC}	2.50 ms	0.747 ms
ECC point Addition	T_{ECA}	0.134 ms	0.003 ms
Hash Function (16 bytes)	$T_{HF} \approx T_{AHF}$	0.345 ms	0.060 ms
Physical Unclonable Function (PUF)	T_{PUF}	0.5 μs	-
ASCON (AEAD scheme)	T_{AC}	0.35 ms	0.061 ms
FE bio-metric key reproduction	$T_{Rep} \approx T_{ECC}$	2.50 ms	0.747 ms

C. Scyther-Based Security Analysis

A formal verification tool Scyther [35] is utilized to study the design defects and characteristics of RACP-SG. Scyther uses a security protocol description language (SPDL), a python-like language, for the implementation of the proposed security protocol, which applies the semantics given in [35]. For a security protocol defined in SPDL, Scyther tries to achieve: 1) affirmation of the claims defined in the specified protocol; 2) affirmation of the security claims that are generated automatically by the Scyther tool; and 3) the comprehensive characterization of the defined roles. There are two roles described in RACP-SG, namely, SM_n and SEP_j. It is explicit from Fig. 5 that RACP-SG is secure. Fig. 5 also explicates that both the manually (defined in SPDL) presented claims in the SPDL script, i.e., Claim (SM, Secret, SK) and Claim (SEP, Secret, SK), and the automatically produced claims, i.e., weak agreement, aliveness, and noninjective agreement (niagree), are "OK."

VII. PERFORMANCE EVALUATION

In this section, we compare RACP-SG with the schemes presented in [13], [14], and [36]–[38]. For performance metrics, we consider security capabilities and communication, storage, and computational overheads. We simulate SM using Ubuntu LTS-16.4 and Raspberry Pi-3/Quad core @1.2 GHz with 1-GB RAM and SEP using Ubuntu LTS-16.4 and Intel Core i7-6700 CPU @ 3.40G with 8-GB RAM. Furthermore, we use ASCON and PyCrypto—a python-based cryptography library—to find out the execution time of ASCON and various cryptographic primitives. We use Raspberry Pi-3/Quad core @1.2 GHz, and 1-GB RAM to evaluate the computational overhead of PUF like [39]. Table IV presents the computational time of various cryptographic primitives.

TABLE V
SECURITY FEATURES COMPARISON

Features	Bera *et al.* [14]	Das *et al* [36]	Malani *et al.* [37]	Bera *et al* [13]	Bera *et al* [38]	RACP-SG
PI	✓	✓	✓	✓	✓	✓
DI	✓	×	×	✓	×	✓
RA	✓	×	✓	✓	×	✓
MITM	✓	×	✓	✓	×	✓
DS	×	✓	✓	×	✓	✓
UT	✓	✓	×	×	×	✓
Scyther/AVISPA	✓	✓	✓	✓	✓	✓
DCA	✓	✓	✓	✓	✓	✓
DoS	✓	✓	✓	✓	✓	✓
ROM	✓	✓	✓	✓	✓	✓

Note: UT: Anonymity/Untraceablity; SI: Scyther Implantation; MA: Mutual Authentication; RA: Replay Attack; ROM: Random Oracle Model; ✓: denotes the availability of features; × : indicates the feature not available

A. Security Features Comparison

Table V presents the comparison of the security features of RACP-SG and the related AC protocols. It is obvious from the table that the scheme presented in [13] does not provide resistance against the De-Syn attack. Similarly, the scheme presented in [36] cannot protect replay, MITM, and device impersonation attacks and also does not provide untraceability feature. Likewise, the scheme presented in [37] cannot protect the device impersonation attack and untraceability feature. Besides, the scheme presented in [13] is insecure against the De-Syn attack and lacks untraceability feature. Lastly, the scheme presented in [38] cannot protect replay, device impersonation, and MITM attacks and does not render the untraceability feature. On the contrary, RACP-SG is secure against the De-Syn, MITM, replay, and device impersonation attacks while ensuring the anonymity and untraceability of the involved entities in the AC process.

B. Computational Overhead

We evaluate the computational overhead of RACP-SG and the related AC protocol using the computational time of different cryptographic primitives as tabulated in Table IV. The average computational time of each cryptographic primitive is determined after 100 runs. RACP-SG requires $8T_{\text{HF}} + 3T_{\text{ECC}} + 5T_{\text{AC}} + T_{\text{Rep}} + T_{\text{PUF}} \approx 12.87$ ms computational overhead to accomplish the AC process, while Bera *et al.* [14], Das *et al.* [36], Malani *et al.* [37], Bera *et al.* [13], and Bera *et al.* [38] required $18T_{\text{HF}} + 10T_{\text{ECC}} + 3T_{\text{ECA}} \approx 23.037$ ms, $12T_{\text{HF}} + 14T_{\text{ECC}} + 6T_{\text{ECA}} \approx 28.75$ ms, $15T_{\text{HF}} + 12T_{\text{ECC}} + 4T_{\text{ECA}} \approx 25.137$ ms, $10T_{\text{HF}} + 8T_{\text{ECC}} + 2T_{\text{ECA}} + 2T_{\text{PO}} \approx 17.8$ ms, and $22T_{\text{HF}} + 8T_{\text{ECC}} + 2T_{\text{ECA}} \approx 23.41$ ms, respectively. Similarly, Table VI demonstrates that RACP-SG incurs less computational overhead as compared to the related stat of the art. Besides, Fig. 6 shows that RACP-SG requires lesser computational overhead at SEP_j than the existing AC protocols as the number of request increases.

C. Communication Overhead

To estimate the communication overhead, we consider the number of messages exchanged and bits transmitted over the communication channel between SM_n and SEP_j. In addition, the communication overhead is determined by considering the size of various parameters, such as ID_{SM_n}, APauth, timestamp, hash function, random number, and ECC point as 128, 128, 32, 256, 128, and 320 bits, respectively. Two message

TABLE VI
COMPARISON OF COMPUTATIONAL OVERHEAD

Protocol/Scheme	SM/D_a Side	SEP/D_b Side	Total Time
Bera *et al.* [14]	$9T_{HF} + 4T_{ECC} + T_{ECA}$	$9T_{HF} + 6T_{ECC} + 2T_{ECA}$	$18T_{HF} + 10T_{ECC} + 3T_{ECA} \approx$ **23.037 ms**
Das *et al.* [36]	$6T_{HF} + 7T_{ECC} + 3T_{ECA}$	$6T_{HF} + 7T_{ECC} + 3T_{ECA}$	$12T_{HF} + 14T_{ECC} + 6T_{ECA} \approx$ **28.75 ms**
Malani *et al.* [37]	$7T_{HF} + 5T_{ECC} + 2T_{ECA}$	$8T_{HF} + 7T_{ECC} + 2T_{ECA}$	$15T_{HF} + 12T_{ECC} + 4T_{ECA} \approx$ **25.137 ms**
Bera *et al.* [13]	$5T_{HF} + 4T_{ECC} + T_{ECA} + T_{PO}$	$5T_{HF} + 4T_{ECC} + T_{ECA}$	$10T_{HF} + 8T_{ECC} + 2T_{ECA} \approx$ **17.8 ms**
Bera *et al.* [38]	$11T_{HF} + 4T_{ECC} + T_{ECA}$	$11T_{HF} + 4T_{ECC} + T_{ECA}$	$22T_{HF} + 8T_{ECC} + 2T_{ECA} \approx$ **23.41 ms**
RACP-SG	$4T_{AHF} + 2T_{AC} + 2T_{ECC} + T_{Rep} + T_{PUF}$	$4T_{AHF} + T_{ECC} + 3T_{AC}$	$8T_{AHF} + 3T_{ECC} + 5T_{AC} + T_{Rep} + T_{PUF} \approx$ **12.870 ms**

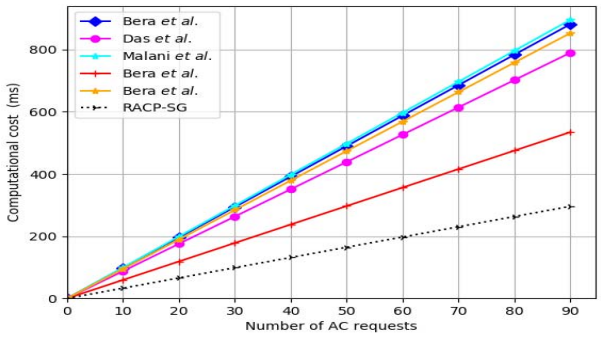

Fig. 6. Computational overhead required to process AC requests from multiple SM_n concurrently.

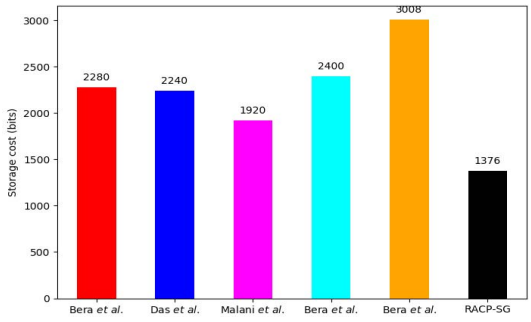

Fig. 7. Total storage overhead required to accomplish the AC process.

TABLE VII
COMMUNICATION OVERHEAD DURING THE AC PHASE

AC Protocol	Messages Exchanged during AC phase	Total (bits)
Bera *et al.* [14]	$SM_n/D_a \xrightarrow{928} SEP_j/D_b \xrightarrow{1122} SM_n/D_a \xrightarrow{288} SEP_j/D_b$	2336
Das *et al.* [36]	$SM_n/D_a \xrightarrow{1472} SEP_j/D_b \xrightarrow{1632} SM_n/D_a \xrightarrow{192} SEP_j/D_b$	3296
Malani *et al.* [37]	$SM_n/D_a \xrightarrow{992} SEP_j/D_b \xrightarrow{1152} SM_n/D_a$	2144
Bera *et al.* [13]	$SM_n/D_a \xrightarrow{672} SEP_j/D_b \xrightarrow{832} SM_n/D_a \xrightarrow{192} SEP_j/D_b$	1696
Bera *et al.* [38]	$SM_n/D_a \xrightarrow{1184} SEP_j/D_b \xrightarrow{1280} SM_n/D_a \xrightarrow{288} SEP_j/D_b \xrightarrow{288} SM_n/D_a$	3040
RACP-SG	$SM_n/D_a \xrightarrow{864} SEP_j/D_b \xrightarrow{544} SM_n/D_b$	1408

exchanges are required to accomplish the AC phase. Message $MG_1 : \{TM_x, SID_{SM_n}, CT_x, APauth_x, Pbk_{SM_n}\}$ sent by SM_n consumes $\{32 + 256 + 128 + 128 + 320\} = 864$ bits and message $MG_2 : \{TM_z, CT_z, APauth_z\}$ transmitted by the SEP_j consumes $\{32 + 256 + 128 + 128\} = 544$ bits. Hence, the total communication overhead of the proposed RACP-SG is $\{864 + 544\} = 1408$ bits. However, the communication overhead of Bera *et al.* [14], Das *et al.* [36], Malani *et al.* [37], Bera *et al.* [13], and Bera *et al.* [38] is 2336, 3296, 2144, 1696, and 3040 bits, respectively. It is evident from Table VII that RACP-SG entails less communication overhead as opposed to the existing AC protocols.

D. Storage Overhead

In the proposed RACP-SG, SM_n and SEP_j need to store $\{SID_{SM_n}, Ch_{SM_n}, RP\}$ and $\{PID_{SM_n}, CT_{SEP_j}, APauth_{SEP_j}, SK_{SEP_j}, Pbk_{SEP_j}\}$, respectively. The storage needs at SM_n and SEP_j are $\{256+128+160\} = 544$ bits and $\{128+256+128+320\} = 832$ bits, respectively. Thus, the total storage requirement of RACP-SG is $\{544 + 832\} = 1376$ bits. However, the storage overhead of Bera *et al.* [14], Das *et al.* [36], Malani *et al.* [37], Bera *et al.* [13], and Bera *et al.* [38] is 2280, 2240, 1920, 2400, and 3008 bits, respectively. Fig. 7 shows the storage overhead comparison of RACP-SG and the related state of the art. It is clear from the figure that RACP-SG requires less storage overhead as compared to the related AC protocols.

VIII. CONCLUSION

This article has proposed a novel AC protocol for the SG systems, called RACP-SG, which employs an LWC-based AEAD scheme and a hash function along with ECC to perform the AC phase. RACP-SG allows an SM and a SEP to mutually authenticate each other and establish an SK that the SM can leverage to communicate with the SEP for data transfer securely. We verified the security of the SK using the widely accepted ROM. By conducting Scyther-based and informal security analyses, we demonstrated that RACP-SG is secure against various covert attacks with reduced storage, communication, and computational overheads compared to the state of the art.

REFERENCES

[1] J. Srinivas, A. K. Das, X. Li, M. K. Khan, and M. Jo, "Designing anonymous signature-based authenticated key exchange scheme for Internet of Things-enabled smart grid systems," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 4425–4436, Jul. 2021.

[2] F. Wu, L. Xu, X. Li, S. Kumari, M. Karuppiah, and M. S. Obaidat, "A lightweight and provably secure key agreement system for a smart grid with elliptic curve cryptography," *IEEE Syst. J.*, vol. 13, no. 3, pp. 2830–2838, Sep. 2019.

[3] M. Z. Gunduz and R. Das, "Cyber-security on smart grid: Threats and potential solutions," *Comput. Netw.*, vol. 169, Mar. 2020, Art. no. 107094.

[4] P. Zhuang, T. Zamir, and H. Liang, "Blockchain for cybersecurity in smart grid: A comprehensive survey," *IEEE Trans. Ind. Informat.*, vol. 17, no. 1, pp. 3–19, Jan. 2021.

[5] A. Triantafyllou, J. A. P. Jimenez, A. D. R. Torres, T. Lagkas, K. Rantos, and P. Sarigiannidis, "The challenges of privacy and access control as key perspectives for the future electric smart grid," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 1934–1960, 2020.

[6] A. Fotovvat, G. M. E. Rahman, S. S. Vedaei, and K. A. Wahid, "Comparative performance analysis of lightweight cryptography algorithms for IoT sensor nodes," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 8279–8290, May 2021.

[7] V. Odelu, A. K. Das, M. Wazid, and M. Conti, "Provably secure authenticated key agreement scheme for smart grid," *IEEE Trans. Smart Grid*, vol. 9, no. 3, pp. 1900–1910, May 2018.

[8] X. Li, F. Wu, S. Kumari, L. Xu, A. K. Sangaiah, and K.-K. R. Choo, "A provably secure and anonymous message authentication scheme for smart grids," *J. Parallel Distrib. Comput.*, vol. 132, pp. 242–249, Oct. 2019.

[9] Y. Chen, J.-F. Martínez, P. Castillejo, and L. López, "A bilinear map pairing based authentication scheme for smart grid communications: Pauth," *IEEE Access*, vol. 7, pp. 22633–22643, 2019.

[10] T.-Y. Wu, Y.-Q. Lee, C.-M. Chen, Y. Tian, and N. A. Al-Nabhan, "An enhanced pairing-based authentication scheme for smart grid communications," *J. Ambient Intell. Hum. Comput.*, to be published. [Online]. Available: https://doi.org/10.1007/s12652-020-02740-2

[11] P. Kumar, A. Gurtov, M. Sain, A. Martin, and P. H. Ha, "Lightweight authentication and key agreement for smart metering in smart energy networks," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 4349–4359, Jul. 2019.

[12] K. Yahya, S. A. Chaudhry, and F. Al-Turjman, "On the security of an authentication scheme for smart metering infrastructure," in *Proc. Emerg. Technol. Comput. Commun. Electron. (ETCCE)*, 2020, pp. 1–6.

[13] B. Bera, D. Chattaraj, and A. K. Das, "Designing secure blockchain-based access control scheme in IoT-enabled Internet of Drones deployment," *Comput. Commun.*, vol. 153, pp. 229–249, Mar. 2020.

[14] B. Bera, A. K. Das, S. Garg, M. J. Piran, and M. S. Hossain, "Access control protocol for battlefield surveillance in drone-assisted IoT environment," *IEEE Internet Things J.*, early access, Jan. 4, 2021, doi: 10.1109/JIOT.2020.3049003.

[15] S. A. Chaudhry, K. Yahya, M. Karuppiah, R. Kharel, A. K. Bashir, and Y. B. Zikria, "GCACS-IoD: A certificate based generic access control scheme for Internet of Drones," *Comput. Netw.*, vol. 191, May 2021, Art. no. 107999.

[16] K. Mahmood, S. A. Chaudhry, H. Naqvi, T. Shon, and H. F. Ahmad, "A lightweight message authentication scheme for smart grid communications in power sector," *Comput. Elect. Eng.*, vol. 52, pp. 114–124, May 2016.

[17] K. Mahmood, S. A. Chaudhry, H. Naqvi, S. Kumari, X. Li, and A. K. Sangaiah, "An elliptic curve cryptography based lightweight authentication scheme for smart grid communication," *Future Gener. Comput. Syst.*, vol. 81, pp. 557–565, Apr. 2018.

[18] D. Abbasinezhad-Mood and M. Nikooghadam, "An anonymous ECC-based self-certified key distribution scheme for the smart grid," *IEEE Trans. Ind. Electron.*, vol. 65, no. 10, pp. 7996–8004, Oct. 2018.

[19] M. F. Ayub, M. A. Saleem, I. Altaf, K. Mahmood, and S. Kumari, "Fuzzy extraction and PUF based three party authentication protocol using USB as mass storage device," *J. Inf. Security Appl.*, vol. 55, Dec. 2020, Art. no. 102585.

[20] M. Tanveer, G. Abbas, and Z. H. Abbas, "LAS-6LE: A lightweight authentication scheme for 6LoWPAN environments," in *Proc. 14th Int. Conf. Open Source Syst. Technol. (ICOSST)*, Lahore, Pakistan, 2020, pp. 1–6.

[21] S. A. Chaudhry, H. Alhakami, A. Baz, and F. Al-Turjman, "Securing demand response management: A certificate-based access control in smart grid edge computing infrastructure," *IEEE Access*, vol. 8, pp. 101235–101243, 2020.

[22] H. J. Jo, I. S. Kim, and D. H. Lee, "Efficient and privacy-preserving metering protocols for smart grid systems," *IEEE Trans. Smart Grid*, vol. 7, no. 3, pp. 1732–1742, May 2016.

[23] K. Mahmood *et al.*, "Pairing based anonymous and secure key agreement protocol for smart grid edge computing infrastructure," *Future Gener. Comput. Syst.*, vol. 88, pp. 491–500, Nov. 2018.

[24] X.-C. Liang, T.-Y. Wu, Y.-Q. Lee, C.-M. Chen, and J.-H. Yeh, "Cryptanalysis of a pairing-based anonymous key agreement scheme for smart grid," in *Advances in Intelligent Information Hiding and Multimedia Signal Processing*. Cham, Switzerland: Springer, 2020, pp. 125–131.

[25] L. Wu, J. Wang, S. Zeadally, and D. He, "Anonymous and efficient message authentication scheme for smart grid," *Security Commun. Netw.*, vol. 2019, May 2019, Art. no. 4836016.

[26] H. M. S. Badar, S. Qadri, S. Shamshad, M. F. Ayub, K. Mahmood, and N. Kumar, "An identity based authentication protocol for smart grid environment using physical uncloneable function," *IEEE Trans. Smart Grid*, vol. 12, no. 5, pp. 4426–4434, Sep. 2021.

[27] A. A. Khan, V. Kumar, M. Ahmad, and S. Rana, "LAKAF: Lightweight authentication and key agreement framework for smart grid network," *J. Syst. Archit.*, vol. 116, Jun. 2021, Art. no. 102053.

[28] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[29] C. Dobraunig, M. Eichlseder, F. Mendel, and M. Schläffer. (2016). *ASCON Lightweight Authenticated Encryption & Hashing*. [Online]. Available: http://ascon.iaik.tugraz.at

[30] M. Tanveer, A. H. Zahid, M. Ahmad, A. Baz, and H. Alhakami, "LAKE-IoD: Lightweight authenticated key exchange protocol for the Internet of Drone environment," *IEEE Access*, vol. 8, pp. 155645–155659, 2020.

[31] M. Tanveer, G. Abbas, Z. H. Abbas, M. Waqas, F. Muhammad, and S. Kim, "S6AE: Securing 6LoWPAN using authenticated encryption scheme," *Sensors*, vol. 20, no. 9, p. 2707, 2020.

[32] M. Tanveer, A. U. Khan, N. Kumar, and M. M. Hassan, "RAMP-IoD: A robust authenticated key management protocol for the Internet of Drones," *IEEE Internet Things J.*, early access, Jun. 4, 2021, doi: 10.1109/JIOT.2021.3084946.

[33] F. Abed, C. Forler, and S. Lucks, "General classification of the authenticated encryption schemes for the CAESAR competition," *Comput. Sci. Rev.*, vol. 22, pp. 13–26, Nov. 2016.

[34] M. Tanveer, G. Abbas, Z. H. Abbas, M. Bilal, A. Mukherjee, and K. S. Kwak, "LAKE-6SH: Lightweight user authenticated key exchange for 6LoWPAN-based smart homes," *IEEE Internet Things J.*, early access, Jun. 3, 2021, doi: 10.1109/JIOT.2021.3085595.

[35] C. J. F. Cremers, "The scyther tool: Verification, falsification, and analysis of security protocols," in *Proc. Int. Conf. Comput.-Aided Verif.*, 2008, pp. 414–418.

[36] A. K. Das, M. Wazid, A. R. Yannam, J. J. P. C. Rodrigues, and Y. Park, "Provably secure ECC-based device access control and key agreement protocol for IoT environment," *IEEE Access*, vol. 7, pp. 55382–55397, 2019.

[37] S. Malani, J. Srinivas, A. K. Das, K. Srinathan, and M. Jo, "Certificate-based anonymous device access control scheme for IoT environment," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9762–9773, Dec. 2019.

[38] B. Bera, S. Saha, A. K. Das, and A. V. Vasilakos, "Designing blockchain-based access control protocol in IoT-enabled smart-grid system," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5744–5761, Apr. 2021.

[39] T. Alladi, N. Naren, G. Bansal, V. Chamola, and M. Guizani, "SecAuthUAV: A novel authentication scheme for UAV-ground station and UAV-UAV communication," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 15068–15077, Dec. 2020.